"Express Mail" mailing label number EV214167363US
Date of Deposit July 29, 2003
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" services under 37 C.F.R. 1.10 on the date indicated above and is addressed to the Assistant Commissioner For Patents, Washington, D.C. 20231.
Typed Name of Person Mailing Paper or Fee: Tamra F. Paulin
Signature: _____

PATENT APPLICATION
DOCKET NO. 100203270-1

PREPARING ELECTRONIC DATA FOR TRANSMISSION

INVENTORS:

Kristofer Erik Metz
Nolan W. Letellier

PREPARING ELECTRONIC DATA FOR TRANSMISSION

BACKGROUND

[0001] With a word processor or other desktop publishing application, a user can open or create an electronic document. To print, the application, with use of an operating system, sends instructions to a print driver to assemble a print job for a laser printer or other image forming device. The print driver acts as a translator between the application and the image forming device, transforming generic printing instructions received from the application into device specific instructions – print jobs – capable of being acted upon by the image forming device.

[0002] An electronic document can be made of up of text, line-art, and/or raster data. Text and line art, as compared to raster data, require relatively small amounts of memory. Raster data are pixels selected and arrange to reveal one or more images. For black and white images, each pixel can be represented by a single bit. A high quality, color image can require forty bits to define each pixel. A relatively small 800 x 600 pixel high quality, color image, then, can be 19,200,000 bits or 2.4 megabytes in size.

[0003] Once the print job is assembled, the print driver transmits the print job through a physical computer port to the image forming device. Port examples include wired or wireless Ethernet, USB (Universal Serial Bus), parallel, serial, and infra-red ports. Different ports have different transmit rates. For example a print job sent through an Ethernet port will be delivered more quickly than a print job sent through a serial port. Smaller print jobs are transmitted relatively quickly regardless of the type of port used. However, a larger print job – for example, a job containing a large amount of raster data – can require a noticeable amount of time to be transmitted resulting in a significant delay when transmitted through a slower port.

DESCRIPTION OF THE DRAWINGS

[0004] Fig. 1 illustrates an exemplary computing environment in which embodiments of the present invention can be implemented.

[0005] Fig. 2 is a block diagram that illustrates programming in the computing environment of Fig. 1 according to an embodiment of the present invention.

[0006] Fig. 3 illustrates another exemplary computing environment in which embodiments of the present invention can be implemented.

[0007] Fig. 4 a block diagram that illustrates programming in the computing environment of Fig. 3 according to an embodiment of the present invention.

[0008] Fig. 5 is block diagram further illustrating programming for implementing an embodiment of the present invention.

[0009] Fig. 6 is an exemplary flow diagram illustrating steps taken to practice an embodiment of the present invention.

DETAILED DESCRIPTION

[0010] ***INTRODUCTION:*** To decrease time it takes to transmit a print job to an image forming device, the print job or a portion of the print job can be compressed. However, compressing and decompressing also take time. So long as the time required to compress and decompress does not exceed the time saved by transmitting an at least partially compressed print job, a decrease in overall printing time can be realized. Various embodiments of the present invention operate to compare an expected time delay caused by compressing and decompressing electronic data with the time saved by transmitting the electronic data if compressed. If the time saved is greater than the time to transmit the data uncompressed, then the electronic data is compressed and transmitted. Otherwise, the electronic data is transmitted but not compressed.

[0011] ***COMPONENTS:*** Fig. 1 illustrates an exemplary computing environment 10 in which various embodiments of the present invention may be implemented. Environment 10 includes image forming device 12, computer 14, and print server 16. Image forming device 12 represents any device capable of forming a printed image on a page. Examples include laser printers, ink printers, copiers, facsimile devices, commercial printing presses, and multi-function all-in-one devices. Computer 14 represents any computing device capable of sending a print job to be printed by image forming device 12. A print job is a series of instructions directing image forming device 12 to form physical representation(s) of en electronic

document on one or more pages. The term page includes a sheet of paper and any other print media such as, but not limited to, an envelope or transparency. Print job instructions can include directions to form text, graphics, or a combination of both.

[0012] Image forming device 12 is capable of processing and printing one print job at a time. Print server 16 represents any computing device capable of managing multiple print jobs simultaneously sent to image forming device 12. Link 18 interconnects image forming device 12, computer 14, and print server 16. Link 18 represents generally a cable, wireless, or remote connection via a telecommunication link, an infrared link, a radio frequency link, or any other connector or system of connectors that provide electronic communication between image forming device 12, computer 14, and print server 16. Link 18 may include an intranet, the Internet, or a combination of both. The portion of link 18 connecting computer 14 to print server 16 may be different than the portion of link 18 connecting print server 16 to image forming device 12. For example computer 14 may be connected to print server 16 via an Ethernet connection. Print server 16 may be connected to image forming device via a parallel connection.

[0013] Referring now to Fig. 2, computer 14 includes application 20 and driver 22. Application 20 represents any programming capable of instructing that an electronic document be printed. For example, application 20 may be a word processor or other desktop publishing application. Driver 22 provides programming for translating printing instructions from application 20 into device-specific instructions – a print job – for image forming device 12. Driver 22 is also capable of selectively compressing a print job or portion thereof. For example, where a print job includes text, line art, and raster data, driver 22 can compress the entire print job or selectively compress only the raster data. Not shown is an operating system that facilitates electronic communication between application 20 and driver 22.

[0014] Print server 16 includes queue 24, queue manager 26, port 28, and bandwidth module 30. Queue 24 represents generally a temporary electronic holding bin for print jobs directed to a busy or malfunctioning image forming device 12. Queue manager 26 represents generally any programming capable of

administering print jobs within queue 24. For example, queue manager 26 intercepts print jobs directed to image forming device 12 and places them in queue 24. As image forming device 12 becomes available, queue manger 26 releases the print jobs one by one from queue 24. It is noted that queue manager 26 may also perform the functions of driver 22. For example, application 20 may send printing instructions directly to queue 24. Queue manager 26, then, translates those instructions into a print job for image forming device 12.

[0015] Port 28 represents a physical interface through which print jobs are transmitted from print server 16 en route to image forming device 12. Bandwidth module 30 represents generally any programming capable of supplying data relating a transmit rate at which print jobs are transmitted to image forming device 12. A transmit rate is the rate at which electronic data such as a print job can be sent through port 28. Transmit rates are typically measured in bytes per second. Bandwidth module 30 can identify a transmit rate in a variety of manners. This includes but is not limited to identifying the transmit rate according to the port through which print jobs are transmitted as well as calculating a transmit rate by measuring the time it takes a print job to leave a print queue. While bandwidth module 30 is shown running on server 16, it may instead be found on computer 14 or any other computing device interconnected with computer14 and server 16.

[0016] Fig. 3 illustrates another exemplary computing environment 32 in which embodiments of the present invention may also be implemented. In computing environment 32, computer 14 and image forming device 12 are interconnected by link 18. Print server 16 is not present. Referring now to Fig. 4, computer 14, in addition to application 20 and driver 22, includes queue 24, queue manager 26, port 28, and bandwidth module 30.

[0017] Without regard to their location, Fig. 5 helps to further illustrate driver 22, queue 24, and bandwidth module 30. As an example, queue 24 is holding a number of print jobs 34. Each print job has an identifier 36, a status 38, and a size 40. In the example of Fig. 5, print job (one) is transmitting, meaning that it is being sent to image forming device 12. The other print jobs are queued meaning that each is being held until image forming device 12 is ready to print it.

[0018] Bandwidth module 30 includes port identifier 42 and throughput detector 44. It is noted however that bandwidth module 30, while shown as including both components 42 and 44, need only include one or the other. Port identifier 42 and throughput detector 44 each represent programming capable of supplying driver 22 with a transmit rate for port 28. In doing so, port identifier 42 infers a transmit rate based on the type of port 28. For example, port identifier 42 may identify port 28 as being of a specific type – for example – as being a parallel port, a USB port, an Ethernet port, or an infra-red port. Ports of specific types are capable of known or assumed transmit rates. Once port identifier 42 identifies the type of port 28, a transmit rate can be inferred and supplied to driver 22. Port identifier 42, for example, may be programmed with assumed transmit rates for ports of varying types. By identifying the a port type, port identifier 42 can supply driver 22 with an assumed transmit rate for that port.

[0019] Throughput detector 44 represents generally any programming capable of measuring and supplying the actual transmit rate for port 28 to driver 22. For example, throughput detector 44 may monitor queue 24. Noting when queue manager 26 releases a print job 34, throughput detector 44 measures a time required to transmit that print job. With the size 40 of the print job known, throughput detector 40 can calculate the transmit rate – size divided by time.

[0020] Driver 22 includes translator 46, compressor 48, duration module 50, and compression selector 52. Translator 46 represents generally any programming capable of translating the generic printing instructions received from application 20 into device specific printing instructions – a print job – for image forming device 12.

Compressor 48 represents generally any programming capable of compressing electronic data like a print job or a portion of a print job like raster data. Where compressor 48 compresses raster data, it may do so using JPEG (Joint Photographic Experts Group) compression. JPEG is a lossy compression technique that can reduce raster data file size by up to ninety-five percent.

[0021] Duration module 50 represents generally any programming capable of (1) calculating a duration for compressing and decompressing a print job or portions of a print job; (2) calculating a duration for transmitting the print job if not compressed; (3) and calculating a duration for transmitting the print job if

compressed. Noting a size of the print job, duration module 50 can calculate a compression/decompression time (C/DT) – that is, the time delay caused by compressing before transmitting and then decompressing by image forming device 12 before printing. The duration for decompressing may, for example, be calculated based upon presumed capabilities of image forming device 12. Noting a size of the uncompressed print job and a transmit rate supplied by bandwidth module 30, duration module 50 can calculate an uncompressed transmit time (UTT). Noting the size of the uncompressed print job, duration module 50 can predict a new size of the print job if compressed. Based on the compressed size and the transmit rate, duration module 50 can calculate a compressed transmit time (CTT).

[0022] Compression selector 52 represents generally any programming capable of selectively instructing compressor 48 to either compress or not compress. Compression selector 52 can do so by determining whether the time saved by compressing exceeds a delay caused by compressing and decompressing – in other words – whether UTT-CTT > C/DT. If so, compression selector 52 instructs compressor 48 to compress. Otherwise, compression selector 52 instructs compressor 48 not to compress.

[0023] It is noted that compressor 48, duration module 50, and compression selector 52 may be provided on print server 16 (see Fig. 2) rather than driver 22. As noted above, queue manager 26 may perform the functions of driver 22. Application 20 may send printing instructions directly to queue 24. Queue manager, then, translates those instructions into a print job for image forming device 12.

[0024] The block diagrams of Figs. 2, 4, and 5 show the architecture, functionality, and operation of various embodiments of the present invention. Each block may represent in whole or in part a module, segment, or portion of code that comprises one or more executable instructions to implement the specified logical function(s). Each block may represent a circuit or a number of interconnected circuits to implement the specified logical function(s). It is noted that in Figs. 2 and 4, the operating systems of computer 14 and server 16 are not shown. Those operating systems provide software platforms on top of which application 20, driver

22, queue manager 26, and bandwidth module 30 run. A number of the interactions between the programming elements running on computer 14 and server 16 are facilitated by those operating systems.

[0025] Also, the present invention can be embodied in any computer-readable media for use by or in connection with an instruction execution system such as a computer/processor based system or an ASIC (Application Specific Integrated Circuit) or other system that can fetch or obtain the logic from computer-readable media and execute the instructions contained therein. "Computer-readable media" can be any media that can contain, store, or maintain programs and data for use by or in connection with the instruction execution system. Computer readable media can comprise any one of many physical media such as, for example, electronic, magnetic, optical, electromagnetic, infrared, or semiconductor media. More specific examples of suitable computer-readable media include, but are not limited to, a portable magnetic computer diskette such as floppy diskettes or hard drives, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory, or a portable compact disc.

[0026] ***OPERATION:*** The operation of embodiments of the present invention will now be described with reference to the exemplary flow diagram of Fig. 6. Electronic data is received (step 54). For example, driver 22 receives printing instructions from application 20 and translates the instructions into a print job that includes raster data. A transmit rate is identified (step 56). With reference to Fig. 5, this is accomplished by bandwidth module 30 which may either infer a transmit rate by identifying port 28 or detect an actual transmit rate by monitoring queue 24.

[0027] The time required to compress and later decompress (C/DT) the electronic data is calculated (step 58). Duration module 50 can do so by noting the uncompressed size of the electronic data and the type of compression to be used. For example, compressing and decompressing raster data of a known size using JPEG compression takes a predictable amount of time. A compressed transmit time (CTT) and an uncompressed transmit time (UTT) are calculated (steps 60 and 62). Duration module 50 can do so by noting the uncompressed size of the electronic data, calculating the compressed size of the electronic data, and then

using the transmit rate identified in step 56 to calculate transmit times for both sizes.

[0028] It is then determined whether C/DT + CTT > UTT (step 64). In other words it is determined whether the time required to compress and decompress the electronic data plus the time for transmitting the electronic data if compressed exceeds the time for transmitting the electronic data if not compressed. If it does not, the electronic data is compressed (step 66). Otherwise, the electronic data is not compressed (step 68). In either case, the electronic data is then transmitted (step 70).

[0029] Although the flow chart of Fig. 6 shows a specific order of execution, the order of execution may differ from that which is depicted. For example, the order of execution of two or more blocks may be scrambled relative to the order shown. Also, two or more blocks shown in succession may be executed concurrently or with partial concurrence. All such variations are within the scope of the present invention.

[0030] ***CONCLUSION:*** The present invention has been shown and described with reference to the foregoing exemplary embodiments. It is to be understood, however, that other forms, details, and embodiments may be made without departing from the spirit and scope of the invention which is defined in the following claims.